

# Penerapan Pengenalan Wajah dengan Metode Eigenface dalam Intelligent Home Security

**Akhmad Hendriawan**

*hendri@eepis-its.edu*

*Politeknik Elektronika Negeri Surabaya*

*Institut Teknologi Sepuluh Nopember(ITS) Surabaya.*

*Kampus ITS, Sukolilo, Surabaya 60111*

## Abstrak

Pengenalan wajah adalah salah satu teknologi biometrik yang telah banyak diaplikasikan dalam sistem security selain pengenalan retina mata, sidik jari dan iris mata. Ada beberapa macam metoda pengenalan wajah yang umum digunakan dalam penelitian beberapa diantaranya adalah neural network, jaringan syaraf tiruan, neuro fuzzy adaptif dan eigenface. Secara khusus dalam paper menawarkan metode Eigenface untuk pengenalan wajah secara real-time menggunakan webcam dengan menambahkan metode viola-jones pada deteksi awal wajah. Metode ini mempunyai komputasi yang sederhana dan cepat dibandingkan dengan penggunaan metode yang memerlukan banyak pembelajaran seperti jaringan syaraf tiruan. Secara garis besar proses dari aplikasi ini adalah hasil gambar RGB dari kamera dilakukan proses resize kemudian dilakukan konversi ke gray dan melakukan perataan cahaya dengan menggunakan histogram equalisasi. Hasil pemrosesan tersebut digunakan untuk mendeteksi wajah dengan menggunakan metode viola-jones. Wajah yang berhasil dideteksi kemudian diproses menggunakan metode eigenface. Metode ini berfungsi untuk menghitung eigenvalue dan eigenvector yang akan digunakan sebagai fitur dalam melakukan pengenalan. Metode Euclidean distance digunakan untuk mencari jarak terkecil dari data fitur yang didapat. Dari percobaan dan pengujian yang dilakukan Alat dapat mengenali citra wajah dengan tingkat keberhasilan sampai 87%.

**Katakunci:** eigenface, gambar, pengenalan wajah, keamanan rumah

## 1. Pendahuluan

Sistem keamanan rumah pada saat ini lebih banyak memanfaatkan tenaga manusia dari pada sistem otomatis. Sistem keamanan rumah dengan alarm telah banyak diterapkan untuk mendeteksi adanya pencuri. Tetapi sistem tersebut tidak bisa membedakan antara pemilik rumah dan pencuri, sehingga tingkat keamanannya sangat rendah. Penggunaan pengenalan wajah pada paper ini adalah untuk mengenali wajah pemilik rumah dan membedakannya dengan wajah pencuri. Pengenalan wajah manusia dalam gambaran visual dapat di implementasikan ke dalam banyak aplikasi yang mencakup pengenalan wajah dengan menggunakan metode eigenface, yang mengisyaratkan pemahaman, pengawasan oleh aplikasi tersebut. Konsep dari eigenface adalah serangkaian eigenvektor yang digunakan untuk mengenali wajah manusia dalam suatu *computer vision*. Eigenvektor berasal dari *covariance matrix* yang memiliki distribusi probabilitas yang tinggi dan dimensi ruang vector untuk mengenali kemungkinan sebuah wajah.

## 2. Latar Belakang

Desain Sistem Pengenalan Wajah Dengan Variasi Ekspresi dan Posisi yang pernah dilakukan oleh Prasetyo dan Rahmatun [6] menggunakan metode eigenface. Metode eigenface disini digunakan untuk mengenali wajah dengan berbagai ekspresi dan posisi. Input

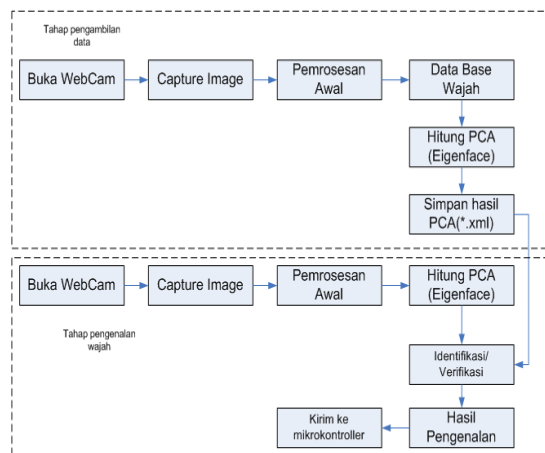
yang diperlukan pada aplikasi ini adalah berupa citra wajah dengan ukuran dan resolusi yang sama. Output aplikasi ini adalah *class* terdekat dari wajah yang ingin dikenali. Penelitian lain dari Ika [3] tentang pengenalan wajah dengan menggunakan metode Principal Component Analysis (PCA) yang digunakan untuk mereduksi dimensi Untuk menghasilkan vector basis orthogonal yang disebut eigenface. Dewi AR [1] memanfaatkan teknologi biometrik pada kehidupan untuk mengidentifikasi dan mengenal karakteristik wajah manusia. Penelitian ini mengembangkan sistem yang memisahkan citra wajah ke dalam komponen wajah, kemudian mengekstraksinya ke dalam fitur mata, hidung, mulut, dan batas wajah pada citra diam tunggal yang diambil dari posisi tampak depan. Antara tiap komponen diukur jaraknya, kemudian dikombinasikan dengan fitur lainnya untuk membentuk semantik wajah. Penelitian dari Jelita [4] yang berjudul Aplikasi Jaringan Syaraf Tiruan Untuk Mengenali Wajah Seseorang merupakan sebuah penelitian yang memerlukan banyak training data. Tapi dalam pengenalan masih banyak terjadi kesalahan seperti yang tertulis dalam batasan masalah. Sistem ini mengambil masukan berupa video dan diharuskan mempunyai background yang sama antara data training dan pelatihan, untuk melakukan background removal. Sistem home security yang dikembangkan oleh Kouma [5] dalam thesisnya menggunakan pengenalan wajah untuk mengenali wajah siapa saja yang tinggal di

rumah. Jika sistem mendeteksi suatu pengganggu, sistem akan mengirim MMS wajah pengganggu kepada pemilik rumah. Sistem akan otomatis mati jika mengenali salah satu wajah pemilik rumah, dan akan aktif ketika semua penghuni meninggalkan rumah.

Kelemahan dari beberapa penelitian diatas adalah system tidak real-time [5], komputasi terlalu berat dan membutuhkan background yang khusus [4], input tidak langsung berasal dari kamera [6]. Pada paper ini mengaplikasikan pengenalan wajah dengan metode eigenface untuk membedakan wajah antara pemilik rumah dan pencuri secara real-time. Pemilik rumah dan pencuri dimasukkan ke dalam *class* yang berbeda sehingga akan lebih mudah dalam pengenalan. Diharapkan dengan aplikasi ini dapat membuat suatu sistem yang handal dan aman.

### 3. Perancangan Sistem

Gambar 1 merupakan blok diagram software untuk melakukan pengenalan wajah yang terdiri dari dua blok yaitu blok pengambilan data dan blok pengenalan wajah.



Gambar 1 Blok diagram system pengenalan wajah

Tahapan sistem blok pengambilan data adalah sebagai berikut :

1. Menangkap citra wajah (*image capturing*) dapat dilakukan secara langsung (*real time*) menggunakan webcam, setelah terdeteksi adanya gambar wajah pada tampilan window dari webcam.
2. Melakukan normalisasi ukuran citra, RGB ke grayscale, histogram equalization untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya, resize untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input.
3. Menyimpan data wajah yang diambil dalam bentuk \*.pgm.
4. Melakukan proses PCA untuk mengutip bagian terpenting dengan metode eigenface

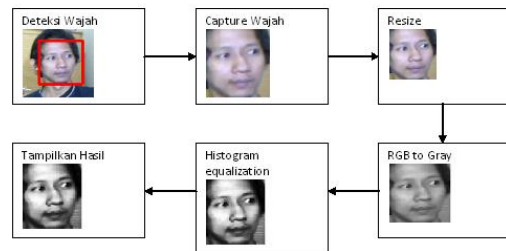
sehingga didapatkan eigenvector dan eigenvalue dari gambar tersebut.

5. Menyimpan citra wajah hasil proses PCA kedalam format xml yang digunakan sebagai database wajah

Sedangkan tahapan proses pengenalan wajah adalah sebagai berikut :

1. Menangkap citra wajah (*image capturing*) dapat dilakukan secara langsung (*real time*) menggunakan webcam, setelah terdeteksi adanya gambar wajah pada tampilan window dari webcam.
2. Melakukan normalisasi ukuran citra, RGB ke grayscale, histogram equalization untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya, resize untuk membuang bagian daerah selain wajah sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input.
3. Melakukan proses PCA untuk mengutip bagian terpenting dengan metode eigenface sehingga didapatkan eigenvector dan eigenvalue dari gambar tersebut.
4. Melakukan proses pengenalan wajah dengan menghitung jarak antara fitur wajah yang ada dalam data dan fitur wajah yang baru. Jarak yang didapat di cari yang terkecil untuk identifikasi.

#### 3.1 Pemrosesan Awal



Gambar 2 Blok diagram sistem pemrosesan awal

Gambar 2 menunjukkan bagaimana proses awal dilakukan dengan tahapan sebagai berikut:

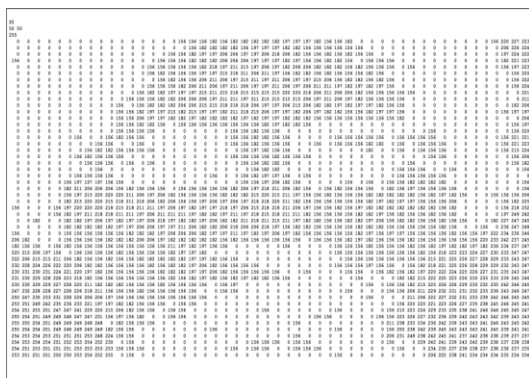
- Mendeteksi citra wajah dengan detektor wajah OpenCV, menggunakan sebuah metoda Viola-Jones.
- menangkap wajah secara otomatis setelah wajah dideteksi.
- Citra wajah yang diperoleh dari berbagai sumber memiliki ukuran yang beragam, karena itu harus diseragamkan sehingga memiliki ukuran 50 x50 piksel.
- Merubah citra wajah kedalam grayscale
- Memperbaiki distribusi nilai derajat keabuan dengan melakukan histogram equalization

#### 3.2 Proses Penyimpanan Data Wajah

Pada proses pengambilan data, jumlah data yang diambil untuk setiap wajah adalah dari 2 pemilik rumah diambil sebanyak masing-masing 5 kali, dengan posisi yang berbeda-beda, jarak yang juga berbeda, kemiringan wajah yang berbeda

pula. Dari setiap pengambilan citra akan dilakukan proses awal seperti yang telah dijelaskan diatas. Kemudian hasil pemrosesan awal akan disimpan dalam format \*.pgm, dikarenakan untuk memudahkan dalam perhitungan eigenvalue dan eigenvector untuk proses pembelajaran. Format \*.pgm adalah sebuah matriks wajah yang berukuran 50 x 50. Matriks yang terdapat dalam fitur wajah pada gambar 3 bernilai antara 0 – 255.

Penyimpanan data wajah ini akan digunakan dalam proses pembelajaran untuk mendapatkan nilai eigenvalue dan eigenvector dari gambar. Setiap nilai dicatat. Kemudian dijumlahkan dan dibagi jumlah berapa kali melakukan percobaan untuk diambil rata rata untuk setiap wajah. Jadi ada 1 nilai hasil rata rata dari 10 kali percobaan untuk setiap wajah pemilik. Nilai inilah yang digunakan sebagai nilai acuan.



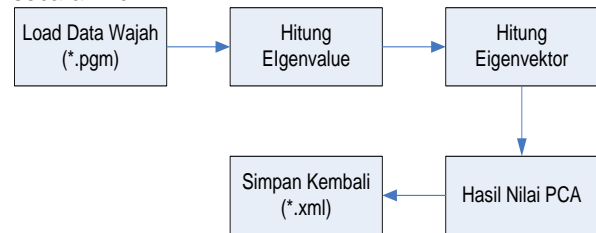
Gambar 3 Wajah dalam bentuk \*.pgm

Setiap pemilik atau penghuni rumah telah disimpan data wajahnya dalam berbagai posisi dan ekspresi, dari frontal menghadap ke depan sampai sudut kemiringan 5° dan disimpan dalam satu class pemilik rumah. Sedangkan dalam class berbeda yaitu class bukan pemilik rumah terdapat banyak data wajah yang diambil secara acak. Jadi wajah yang tersimpan bukan hanya pemilik rumah saja, tetapi wajah bukan pemilik rumah sebagai sample dan pembanding. Pengambilan data wajah dalam class yang berbeda ini berguna untuk proses pengenalan wajah selanjutnya.

### 3.3 Proses Penghitungan Eigenface

Pada langkah ini informasi - informasi yang relevan dari data yang disimpan akan dilakukan tahap penghitungan eigenvalue dan eigenvector baik dari citra wajah keseluruhan maupun dari elemen wajah. Jadi pada tahap ini dilakukan proses lokalisasi elemen wajah. Sifat simetri wajah dapat digunakan membantu proses PCA ini. Eigenvector masing-masing memiliki nilai yang sama seperti aslinya gambar itu sendiri dan dapat dipandang sebagai sebuah gambar. Eigenvector ini merupakan covariance matriks karena itu disebut eigenface. Arah dari eigenvector itu dimana gambar dalam pelatihan ditetapkan berbeda dari gambar yang asli. Secara umum ini akan menjadi langkah komputasi mahal (bila keadaan memungkinkan),

tetapi praktek penerapan eigenface berasal dari kemungkinan untuk menghitung eigenvector yang efisien, tanpa pernah dilakukan komputasi secara rinci.



Gambar 4 Proses penghitungan Eigenface

Proses PCA dilakukan dengan :

1. Mengambil data dari hasil training.
2. Melakukan proses perhitungan dengan menggunakan PCA untuk menemukan *subspace*. Cari matriks covariance :  $S = T^T \times T$
3. Menghitung gambar hasil pelatihan ke dalam PCA *subspace* dengan menggunakan eigenvalue dan eigenvector (eigenface).  $TT^T u_i = \lambda_i u_i$  dan  $Sv_i = T^T T v_i = \lambda_i v_i$ .
4. Mendapatkan nilai eigenface.

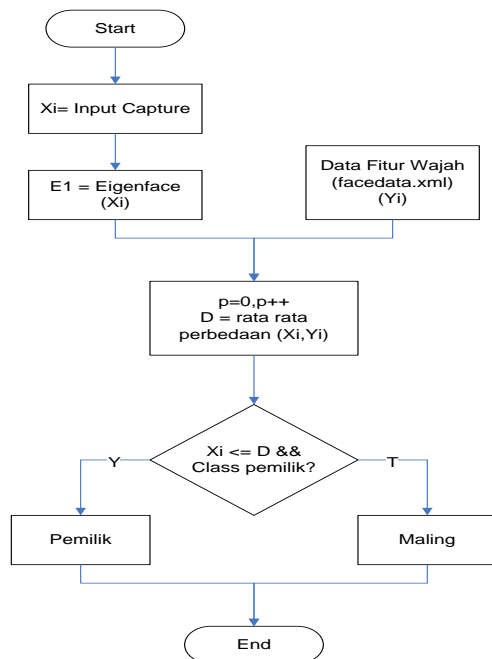
$$\text{Eigenface} = \frac{1}{\sqrt{\text{Eigenvalue}}} * \text{Eigenvector} * T$$

5. Menyimpan semua informasi hasil pelatihan dalam \*.xml.

### 3.4 Proses Pengenalan

Proses pengenalan wajah dilakukan dengan menggabungkan metode Euclidean distance dan k-nearest neighbor. Metode K-nearest neighbor digunakan pada pembuatan class antara pemilik dan pencuri. Diagram alur keseluruhan dalam proses pengenalan wajah dapat dilihat pada gambar 5. Penjelasan proses pengenalan wajah pada gambar 5 sebagai berikut.

1. Mengambil sebuah citra (\*.pgm) .
2. menentukan nilai mean dan eigenvaluennya.
3. Menghitung rata-rata perbedaan antara fitur input (Xi) dan fitur hasil pelatihan (Yi).
4. Jika jarak dari fitur input (Xi) lebih kecil dari pada rata –rata perbedaan dan nilai tersebut masuk kedalam *class* pemilik maka identifikasikan sebagai pemilik.
5. Jika jarak dari fitur input (Xi) lebih besar dari pada rata –rata perbedaan dan berada diluar *class* pemilik maka identifikasikan sebagai pencuri atau bukan pemilik.



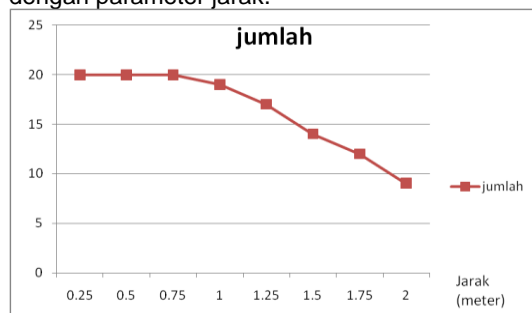
Gambar 5 Proses Pengenalan Wajah

#### 4. Hasil percobaan

Pengujian dilakukan menggunakan webcam logitech quickcam S-5500 dengan resolusi 1,3 megapixel yang dihubungkan dengan komputer intel pentium dualcore dengan clock CPU sebesar 2.66 GHz dengan RAM sebesar 512mb RAM dan OS windows XP servicepack 2

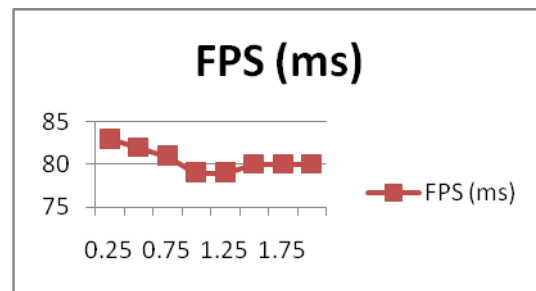
##### 4.1 Pengujian Deteksi Wajah

Pada tahap pengujian deteksi wajah ini akan diuji berdasarkan tingkat ketepatan deteksi wajah dengan parameter jarak.



Gambar 6 Jumlah wajah terdeteksi

Dengan 100 kali ujicoba untuk mendeteksi wajah berhasil terdeteksi sebanyak 92 kali dan 8 kali tidak terdeteksi. Sedangkan gambar selain wajah terdeteksi sebagai wajah sebanyak 3 kali dan 17 kali tidak terdeteksi sebagai wajah. Sehingga tingkat keberhasilan akurasi wajah ini sangat tinggi yaitu mencapai 90,833%.



Gambar 7 Grafik fps terhadap jarak

Pengukuran frame per second ini bertujuan untuk mengetahui kehandalan sistem pendeteksian wajah, yaitu dengan cara mencari tau berapa frame yang dapat diambil dalam waktu satu detik. Contoh :

Jika diketahui nilai fps = 80 ms

Maka frame yang dapat di ambil dalam 1 detik :  
1 detik = 1000 ms,

Jumlah frame =  $1000\text{ms}/80\text{ms} = 12,5$  frame

Jadi dalam waktu satu detik bisa mengambil 12,5 frame, sehingga sistem cukup handal.

##### 4.2 Pengujian Pemrosesan Awal Wajah

Setelah wajah terdeteksi, langkah-langkah selanjutnya adalah proses awal wajah yang meliputi capture wajah, memotong lokasi wajahnya saja dari keseluruhan gambar yang tercapture, konversi ke gray scale, Histogram equalization untuk memperbaiki citra gambar dari noise. Proses awal dilakukan untuk masukan dalam proses penghitungan eigenface sebelum dilakukan proses pengenalan.

Tabel 1 Hasil uji pemrosesan awal

No.	Obyek	Proses sempurna	Proses tidak sempurna
1	Gambar wajah	86 kali	14 kali
2	Gambar selain wajah	1 kali	19 kali

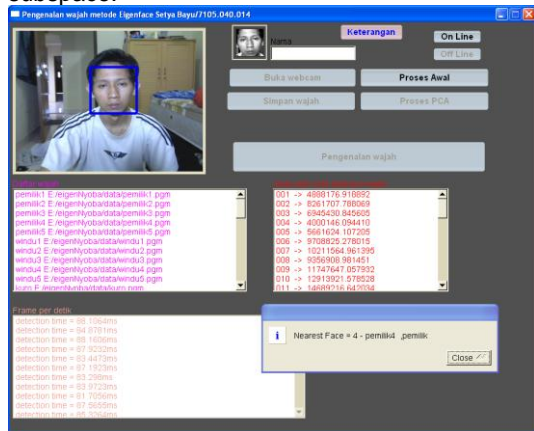
Dari tabel 1 dilakukan pengukuran performa sistem pemrosesan awal wajah, dengan uji coba sebanyak 100 kali dengan gambar wajah maka didapatkan proses sempurna sebanyak 86 kali sedangkan proses tidak sempurna sebanyak 14 kali. Sedangkan performa dengan gambar selain wajah didapatkan proses sempurna hanya 1 kali dari 20 kali pengukuran dan proses tidak sempurna terdapat 19 kali. Sehingga didapatkan akurasi dari pemrosesan awal wajah sebanyak 87,5% . Dengan akurasi yang cukup besar ini maka dapat dikatakan sistem dapat bekerja dengan baik pada tahap pemrosesan awal, karena keberhasilan tahapan ini akan menentukan untuk tahap pengenalan wajah selanjutnya.

##### 4.3 Pengujian Pengenalan Wajah

Dalam pengujian ini dilakukan untuk mengetahui keberhasilan sistem pengenalan wajah. Dengan



data wajah yang telah dilakukan pembelajaran sebelumnya dengan metode eigenface maka tahap pengenalan wajah ini dapat dilakukan dengan menguji gambar wajah dengan mencari gambar pelatihan yang terdekat kedalam PCA subspace.



Gambar 8 Hasil pengenalan wajah

Sistem pengenalan wajah ini digunakan untuk membedakan antara pemilik rumah dengan bukan pemilik. Pengenalan dilakukan dengan memasukkan data pemilik wajah kedalam satu class, masing-masing pemilik 5 data wajah. Sehingga data selain dalam kelompok pemilik diidentifikasi sebagai pencuri. Seperti pada gambar 8 pengenalan menghasilkan bahwa wajah adalah pemilik.

Tabel 2 Hasil uji pengenalan wajah

	Dikenali pemilik	Pencuri
Pemilik 1	87	13
Pemilik 2	89	11
Wajah random	14	86

Akurasi pengenalan :

- Wajah Pemilik 1 : wajah random dikenali pemilik →  $((87+86)/(87+86+13+14)) * 100\% = 86,5\%$
- Wajah Pemilik 2 : wajah random dikenali pemilik →  $((89+86)/(89+86+11+14)) * 100\% = 87,5\%$
- Wajah random : wajah Pemilik 1 & Pemilik 2 dikenali pemilik →  $((13+11+14)/(13+11+14+87+86+89)) * 100\% = 12,66\%$

#### 4. Kesimpulan

Dari hasil pengujian didapatkan kesimpulan sebagai berikut :

- Tinggi rendahnya unsur pencahayaan yang berada di sekitar obyek sangat mempengaruhi proses pendeteksian.
- Jarak antara wajah dengan webcam sangat berpengaruh dalam proses pendeteksian wajah.
- Kombinasi – kombinasi dari metode viola-jones dapat mendeteksi wajah dengan tepat.
- Proses pengenalan akan berjalan dengan baik bila capture wajah hasil deteksi jelas dan tidak kabur.

- Penyimpanan data hasil pemrosesan awal dalam bentuk \*.pgm untuk memudahkan dalam proses penghitungan PCA.
- Sesuai permasalahan, alat berhasil dapat melakukan pengenalan meskipun posisinya berbedanya. Karena yang digunakan adalah nilai dari eigenface tiap citra wajah yang dibandingkan.
- Tingkat keberhasilan pengenalan wajah menggunakan euclidean distance dan k-Nearest neighbour sangat dipengaruhi oleh deteksi wajah, pemrosesan awal, dan penghitungan dengan PCA(eigenface) sebelumnya.
- Penggunaan class untuk pengelompokan data pemilik rumah dan bukan pemilik rumah sangat efektif digunakan dalam proses verifikasi antara pemilik atau pencuri

#### 5. Pustaka

- [1] A.R. Dewi, "Ekstraksi Fitur dan Segmentasi Wajah Sebagai Semantik Pada Sistem Pengenalan Wajah", Makalah Skripsi Universitas Gunadarma, Jakarta, 2005.
- [2] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," IEEE Trans. PAMI, vol. 20, pp. 23–38, Madison, USA, January 1998
- [3] Ika made, "Face Recognition dengan Menggunakan Metode Principal Component Analysis (PCA)", Makalah Skripsi Universitas Gunadarma, Jakarta, 2003.
- [4] Jelita Raisa, "Aplikasi Jaringan Syaraf Tiruan Untuk Mengenali Wajah Seseorang", Tugas Akhir Politeknik Elektronika Negeri Surabaya Institut Sepuluh Nopember, Surabaya, 2008.
- [5] Kouma Jean-Paul, "Intelligent home security system", Tesis Umeå university, Swedia, 2006.
- [6] Prasetyo Eri, Rahmatun Isna, "Desain Variasi Wajah dengan Variasi Ekspresi dan Posisi Menggunakan Metode Eigenface", Makalah Skripsi Universitas Gunadarma, Jakarta, 2005.
- [7] Rinaldi Munir, "Pengolahan Citra Digital dengan Pendekatan Algoritmik", Penerbit Informatika, Bandung, 2004.
- [10] Bluekid, (<http://derindelimavi.blogspot.com/2008/05/y-z-tanma-2.html>)
- [11] Romdhani s. "Face Recognition Using Principal Component Analysis", [http://eeapp.elec.gla.ac.uk/~romdhani/pca\\_doc/pca\\_doc\\_toc.htm](http://eeapp.elec.gla.ac.uk/~romdhani/pca_doc/pca_doc_toc.htm)
- [12] Seeing with OpenCv, ([http://www.cognotics.com/opencv/servo\\_20\\_07\\_series/part\\_5/index.htm](http://www.cognotics.com/opencv/servo_20_07_series/part_5/index.htm))